SCfM_*PROD*: A SOFTWARE PRODUCT CERTIFICATION MODEL

Jamaiah Haji Yahaya
Faculty of Information Science and Technology,
University Kebangsaan Malaysia,
Bangi, 43650 Selangor, Malaysia
jhyahaya@yahoo.com

Faculty of Information Technology
University Utara Malaysia
Sintok, 06010 Kedah, Malaysia
jamaiah@uum.edu.my

Aziz Deraman, Abdul Razak Hamdan
Faculty of Information Science and Technology,
University Kebangsaan Malaysia,
Bangi, 43650 Selangor, Malaysia
{ad, arh}@ftsm.ukm.my

Abstract—In recent years, there has been an increasing interest in software development and ICT as they become a central issue for business surviving. Companies and software houses are competing to produce software which are claimed to be good and fulfil user's expectation. Therefore, quality aspect of a software product has seen as an important issue but companies that develop the software could not justify and guarantee the quality of their products, thus leaving users in uncertainties. Previous studies have indicated that assessment and certification by an independent assessor may help determining confident in the product. The certification reports are beneficial not only to the users and stakeholders but also to the developers and suppliers. This research proposes a software certification model by end product quality approach or SCfM_*Prod*. It is an alternative approach to monitor and guarantee level of quality of a software product. This model consists of four main entities: pragmatic quality factor (PQF), assessment team, certification representation method and repository. This paper explains in detail of this model

Keywords-Software Certification; software quality; software assessment;software measurements.

I. INTRODUCTION

Many complaints have been reported on the quality of software supplied by vendors or suppliers to users over the years. Some of these have been registered by the CIO of US Department of Defense, the publisher of CIO Magazine and the President of ACM. They complained that software has been degenerating steadily [1]. Vendors are accused of delivering software with bugs that need to be fixed. This is consistent with observations in local companies and organizations in Malaysia. The prevalence of this practice leads to a general perception among clients that the software industry in the country as a whole lacks standards and mechanisms for monitoring or ensuring product quality.

Problems concerning recent circumstances in software assessment show that lacking of mechanism and standard of software quality does not concern on off-the-shelf product but for in-house development products as well. Users and stakeholders are left with reservation and doubt on the reputation and standard of the software being used in their environment and organisation. Findings from previous empirical study conducted in Malaysia agreed that with certification embedded and granted to the software might resolve the uncertainties and suspicion on the status and standing of software product. The survey too has identified main quality attributes, which are crucial to the proposed certification framework. The survey strengthens the demand for better software quality standards and procedures. Furthermore, software certification and independent body authorized by the authority can be formed to ensure that users will be getting software packages that meet expected and contracted quality standards.

This research participates in solving these problems by proposing a model for assessment and certification of software product. This work is towards improving effort in software product certification process particularly in outlining an approach to assess software product and determine the quality status of specific candidate software product. Moreover, the determining requirements are presented in practical quality factor with an aim at supporting certification process. The proposed pragmatic quality factor is derived from ISO 9126 model with enhanced features and capabilities. The proposed model for software certification has several interesting features which will discuss in detail in this paper.

II. SCFM_PROD: A CERTIFICATION MODEL BY PRODUCT QUALITY APPROACH

The certification by product quality approach is an alternative and acceptable approach of certifying software[2]. This paper concentrates and focuses on development and construction of software certification model by product quality approach. A western analogy says that dirty water can run from clean pipes is believed to be true as a good software development processes do not guarantee the excellent quality of product. Thus, assessment of end product software must be independent from the development process. Previous studies [3, 4] show that code analysis and testing software alone will not guarantee the quality of the product. Empirical study by Lauesen and Younessi (1998) claims, "that 45% of defects could not be detected when relying on code analysis alone." Lauesen and Younessi conclude, "many defects cannot be found through analysis because they reflect tacit or undesirable requirements or can be observed only when the product is being used. Detecting such defects is the purpose of good acceptance testing, and it cannot be replaced with code analysis, no matter how sophisticated."

The proposed software product certification framework named SCfM_*prod* is designed based on the following basis:

i) Assessment by independent body is an advantage to the user by conducting unbiased assessment. The independent certification is believed to be the only approach that user should trust and the demands for it are being heard from both publishers and users [5, 6]. While evaluation by the SQA team in the organization or the owner/users of the product is beneficial because they know well of the software and will reduce the time taken for assessment process. Thus in our approach we investigate the possibility of conducting assessment and certification of software product using collaborative perspective approach between the owner/users of the product, developers and independent assessor. The advantages of this approach compares to other approaches are:-

- eliminate bias assessment and evaluation of the product by including independent assessor in the team,
- remove unfairness evaluation by including the owner or users of the product to participate in the assessment process ,
- accelerate the process because the team is familiar with the product and its' environment, and
- protect data confidentiality and privacy by only permitting users to have direct access to the software.

ii) The candidate software product is completed software and is operational in certain environment.

iii) The software quality factors apply in this research are derived from the ISO 9126 model with additional characteristics to accommodate other aspects of software quality requirements.

Figure 1 shows the SCfM_*prod* model. It consists of pragmatic quality factor (PQF), product certification repository, certification representation method, and assessment team. Each of these items will be discuss in detail in the following sections.

A. Pragmatic Quality Factor (PQF)

Previous survey [7] indicated that functionality, efficiency, integrity, maintainability and reliability were the main characteristics with high and very high consideration in assessing software products by respondents in Malaysia.

Table 1 compares the result from the survey and the software quality characteristics according to ISO 9126 model. There are four characteristics resulted from the survey that are equivalent to the ISO 9126 characteristics. The characteristics are efficiency, reliability, functionality and maintainability. Integrity is not included in the ISO model but is considered as high consideration by participants in the survey. However portability and usability are not among the favorite high consideration characteristics in this survey but included in the ISO model. Even though usability is not considered as high consideration by the respondents, the mean score is high and almost achieving level high consideration. The PQF considers this analysis. It is anticipated by combining and filtering these two sources as well as other quality models available from literature.

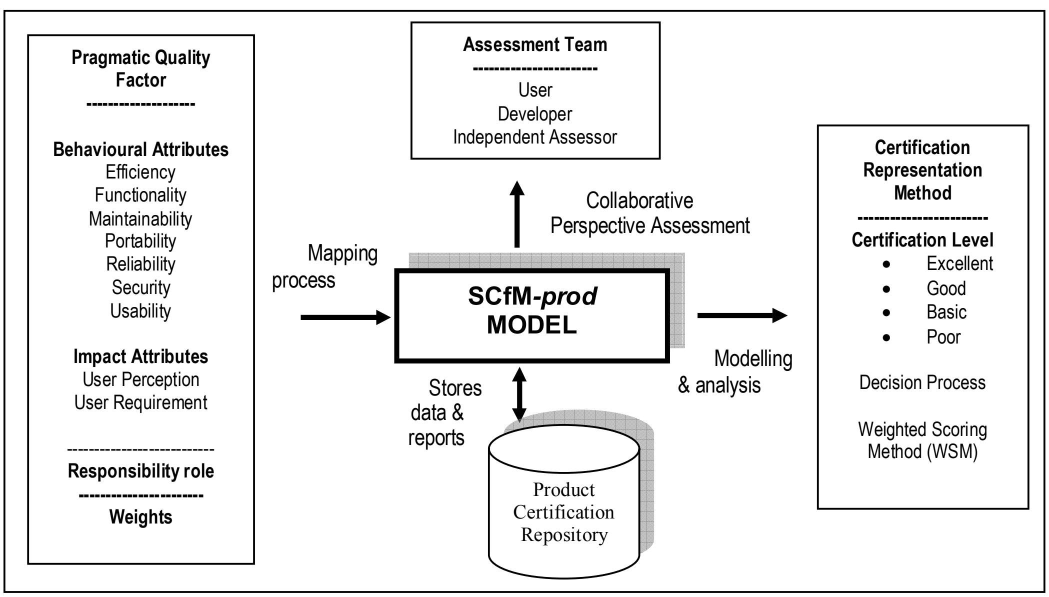

Figure 1. SCfM_*Prod* : Model for software certification by product quality approach

TABLE 1 :
QUALITY CHARACTERISTICS: COMPARISON OF ISO 9126 MODEL AND SURVEY

Quality Characteristics	ISO9126 Model	Survey
Efficiency	x	x
Reliability	x	x
Functionality	x	x
Maintainability	x	x
Portability	x	
Usability	x	
Integrity		x

The PQF is different from other models as we identify factors, subfactors and metrics relevant to requirements for certification process. The metrics are considered relevant if they are measurable at a minimum time of operational. Therefore, there will be some quality metrics that irrelevant for this purpose.

The PQF consists of four distinct components which are the behavioural attributes, the impact attributes, responsibility roles and weights.

1) The Behavioural Attributes

The behavioural attributes include efficiency, functionality, maintainability, portability, reliability, integrity and usability. Each attribute is made up several metrics that shows the measurement aspects of the attributes. The behavioural attribute is defined as a quality feature or characteristics of software. It is derived from ISO 9126 model plus integrity aspect included. In the age of hackers and firewall, the important of integrity aspect has increase [8]. This attribute measure the ability to with-stand attack to its security that comprises of program, data and document. It covers threat and security aspects.

These attributes are decomposed into several sub attributes and then a further level of decompositions to associate with directs measurable metrics. Each sub attributes and metrics are comprises of an information on interviewee's role.

2) The Impact Attributes

The impact attribute defined in PQF refers to the human aspect of quality toward the product. It illustrates the impact of the software in term of quality to the users and also measures the conformity of software to the user requirement. These attributes are important to balance the quality model between technical measurement of software and human factor [9]. Similar to behavioural attributes, the impact attributes are made up of several subattributes and metrics that show the measurement of the attributes. The impact attributes are decomposed into two distinct subattributes, which by means of user perceptions and user requirements. The metrics include measures of popularity, performance, trustworthiness, law and regulation, recommendation, environmental adaptability, satisfaction and user acceptance. Table 2 shows the subattributes and their associated metrics.

TABLE 2: A DECOMPOSITION OF IMPACT ATTRIBUTE

Subattributes	Metric
User Perceptions	*Popularity* *Performance* *Law & Regulation* *Recommendation* *Trustworthiness* *Requirement & Expectation* *Environmental adaptability*
User Requirement	*User acceptance* *Satisfaction*

3) Responsibility Role and Measurement of Metrics

The third component in PQF is the responsibility role. It is defined as the responsibility person to answer the questions related to metrics. It is also named as the interviewee in this model. The PQF has identified specific interviewee to responsible in giving the assessment score of each metrics. The interviewee is the user, developer, independent assessor or combination of these interviewees.

The measurements of metrics used are Likert scale of 1 to 5 based on collaborative perspective among assessment team members. Likert scale is defined as something that we measure the satisfaction based on perception. The Likert technique presents a set of attitude statements. Subjects are asked to express agreement or disagreement of a five-point scale. Each degree of agreement is given a numerical value from one to five. Thus a total numerical value can be calculated from all the responses. The scale used in this approach is recommended to 1 = unacceptable, 2 = below average, 3 = average and 4 = good, 5= excellent.

4) Classification of Attributes and Weight Factors

The weighting factors defined in this model are based on findings from previous empirical study discussed in [10]. We asked respondents in the survey to indicate the levels of consideration which are by means of 1=not considered, 2=low consideration, 3=average, 4=high consideration and 5=very high consideration of all the quality attributes. These criteria are taken into account during assessment exercise of software product in their organizations.

For the purpose of this classification, we are interested to analyse the two modes of considerations that are *Very High Consideration* and *High Consideration* only. Data management and analysis was performed using SPSS and the weight of each attributes is calculated using the following formula:-

$$\text{TotalVH} = \sum_{a=1}^{n} VH_a \qquad (1.1)$$

where n = number of attributes defined in the analysis and VH is the score for Very High Consideration. Then,

$$Weight_a = (VH_a / TotalVH), \quad (1.2)$$

and

$$\% \; Weight_a = (VH_a / TotalVH) * 100 \quad (1.3)$$

where subscript *a* represents an attribute.

From the analysis, the function point approach is used to group and classify attributes into three distinct classifications namely low, moderate and high. Then, the attributes are sorted into these classifications according to the calculated weight score (1.3). The analysis shows that functionality is 14.29% more important compares to other quality attributes defined in this model. It obtains the highest weight in this analysis. Reliability is considered 12.34% more important and integrity is considered 11.69% important. These three attributes (functionality, reliability and integrity) are classified in the classification group of high. Second group of classification defined as moderate includes safety (8.44), efficiency (9.09%), maintainability (7.79%) and usability (7.79%). On the other hand, the third group of classification defined as low includes flexibility (5.84%), Interoperability (6.49%), Intraoperability (5.84%), portability (5.19%) and survivability (5.19%). The classification analysis and method are discussed in detail in [14].

For the purpose of assessment and certification defined in this research we therefore assign weight factor for each group accordingly. This is consistent with the requirements of having different weights for attributes [11]. Table 3 demonstrates the classification of attributes and its weight factor. These factors will be used in our proposed certification model and will be discuss in the next section.

TABLE 3
CLASSIFICATION OF ATTRIBUTES AND WEIGHT FACTORS

Levels	Attributes	Weight Factor
Low	Flexibility Intraoperability Interoperability Portability Survivability	1-4
Moderate	Safety Efficiency Maintainability Usability	5-7
High	Functionality Reliability Integrity	8-10

C. Individual Quality Attributes Assessment

Software quality model defined in this model comprises of attributes, sub attributes, and metrics. The quality score is calculated using the following algorithms and will refer to Table 4. This table represents metrics in individual attribute. M_1, M_2, M_3 and etc represent metrics in specific attributes, A_1, A_2, and etc represent assessor in this model which either user, developer or independent assessor. P_{12}, P_{21}, P_{n1} are the perspective value given by the assessor for each of the metrics.

(4.2)

TABLE 4. TABLE OF METRICS OF ATTRIBUTE

Metrics / *Assessor*	M_1	M_2	M_3	...	M_t
S_1	P_{11}	P_{12}	P_{13}	...	P_{1t}
S_2	P_{21}	P_{22}	P_{23}	...	P_{2t}
•					
•					
S_n	P_{n1}	P_{n2}	P_{n3}	...	P_{nt}
Average (T)					

The average score for each of the metric is calculated as follows: -

$$T_k = \left(\sum_{j=1}^{n} p_{ij} \right) / n \quad , \; k=1,2.....t \quad (1.4)$$

where n represents number of assessor and t represents number of metrics.

Then, the average perspective score (aps) of attribute *a,* is calculated as the following:-

$$aps_a = \left(\sum_{i=1}^{t} T_k \right) / k \quad , \; k=1,2.....t \quad (1.5)$$

Each attribute calculated using formula 1.5 can be used to measure its quality level by :

$$QS_a = \left(\sum_{i=1}^{n} aps_i \; / \, 5 \right) * 100 \quad (1.6)$$

where i = number of assessors. The constant 5 represents the maximum possible value of quality score. The QS score is mapped to a certification representation model to obtain its associate level.

B. The Certification Representation Method

The representation method is elaborated in three main sections which are related to weighted scoring method, certification level and decision process. The discussions are in the ensuing sections.

1) Weighted Scoring Method For Product Assessment

The Weighted Scoring Method (WSM) is typically applied in the following fashion: attributes and sub-attributes (SA) are defined, for each sub-attributes, there are several metrics to measure attributes. Each attribute holds an average perception scale (aps) given by the assessment team and a weight factor. The weight factors are assigned by owner of the product based on its organization requirements and expectations and guided by our model defined in Table 3. The quality score (QS) and percentage quality score (QSP) are formulated as below:-

$$TotW = \left(\sum_{i=1}^{n} w_i \right) \quad (1.7)$$

where w represents weight of attribute i and n is number of attributes.

$$QS_i = (W_i / TotW) * aps_i \quad (1.8)$$

where w represents weight of attribute i and aps_i is score obtain by attribute i. *Aps* is obtained using formula 1.5. Then, to calculate the percentage of quality score obtained for each attribute:-

$$QSP_i = (QS_i / 5) * 100 \quad (1.9)$$

In order to calculate the total quality score of candidate product (the behavioural attributes),

$$TQP = (\sum_{i=1}^{n} QSP_i) \quad (1.10)$$

where QSP represents quality score of attribute i and n is the number of attributes.

TQP score is then mapped to a certification representation model to obtain its associate level.

2) Certification Level

The certification levels are identified and characterised in four distinct levels which are excellent, good, basic and poor. The certification level of product is determined by comparing the score value obtained in equation 1.10. For TQP value greater than 90% and less than 100%, the product obtains a certification level of *excellent*. This means that the software product satisfies all quality criteria and achieves quality level of excellent and satisfactory. Whilst if the TQP score is greater than or equal to 75% and less than 90, the product is classified as *good* which means that it satisfies the quality level of good. If the product gains TQP score greater and equal to 50 and less than 75, the product is identified as *basic* which means that the software satisfies the quality level of basic or average and acceptable. If the TQP score obtains less than 50, the product is identified as poor and unsatisfactory. The classification level is shown in Table 5. The similar classification technique is used in [12].

3) *Decision Process*

Data collected goes through the process of modelling and analyzing using certification representation method. Decision process defined in this model is used as a guideline in the assessment and certification process. Figure 2 illustrates the overall software certification decision process which comprises of the identification process, assessment, retrieving and checking from previous data and award certification status. The process starts by following a path from start that connects activities (boxes) and decisions (diamonds) to submit accreditation. Each activity in the process may enter and exit in several ways. The detail decision process is documented separately [13] and it is not explained in detail in this paper.

C. Product Certification Repository

SCM Repository stores data and reports of software product certification candidates. The SCM design is beneficial for the basis of certification repository architecture.

TABLE 5:
RANKING OF CERTIFICATION LEVELS

TQP Score (q)	Certification Level	Description
90<= q<=100	Excellent	Software satisfies all quality criteria and achieves quality level of excellent.
75<= q < 90	Good	Software satisfies and achieves the quality level of good.
50< =q < 75	Basic	Software satisfies and achieves the quality level of basic which also means average and acceptable.
0 <= q< 50	Poor	Software attains quality level of poor and unsatisfactory.

D. The Assessment Team

The last entity of SCfM_p*rod* model is the assessment team. This model applies a collaborative perspective assessment between user, developer and independent assessor. This assessment technique was discussed in earlier section that elaborated the advantages of this technique.

III. EVALUATION OF THE MODEL

In undertaking the development of a software certification model, there is a need to verify as well as to evaluate its feasibility. To evaluate this model, three care studies were launched collaboratively with three large organizations in Malaysia. Upon completion of the data analysis of the case studies which involve data and reports on assessment and certification using SCfM_*Prod* model, meetings were to obtain feedback from the target organisations. The meeting was setup to present the findings and results of the case studies by the independent assessor or the assessment team leader, to obtain feedback from the cases regarding the integrity of this model and to recommend to the committee future works related to this certification exercise in specific organisations.

The encouraging discussions were conducted with a fruitful source of feedbacks and suggestions from the committees and assessment team. Some of the core highlighted feedbacks and suggestions are: -

a) The committee agrees that this software certification model is a valuable model to assess system operational in the business environment. This model can be used multiple times to monitor the performance of the system during it life span. It provides beneficial information to the developers, owners as well as the stakeholders on the quality status of the system.

b) The weights assigned for individual behavioural attributes in this model are useful for reflecting the business requirements. The committee approves that the validity of the weight values associated with quality attributes defined in this model is depended on the maturity of the person in charge in assigning the values.

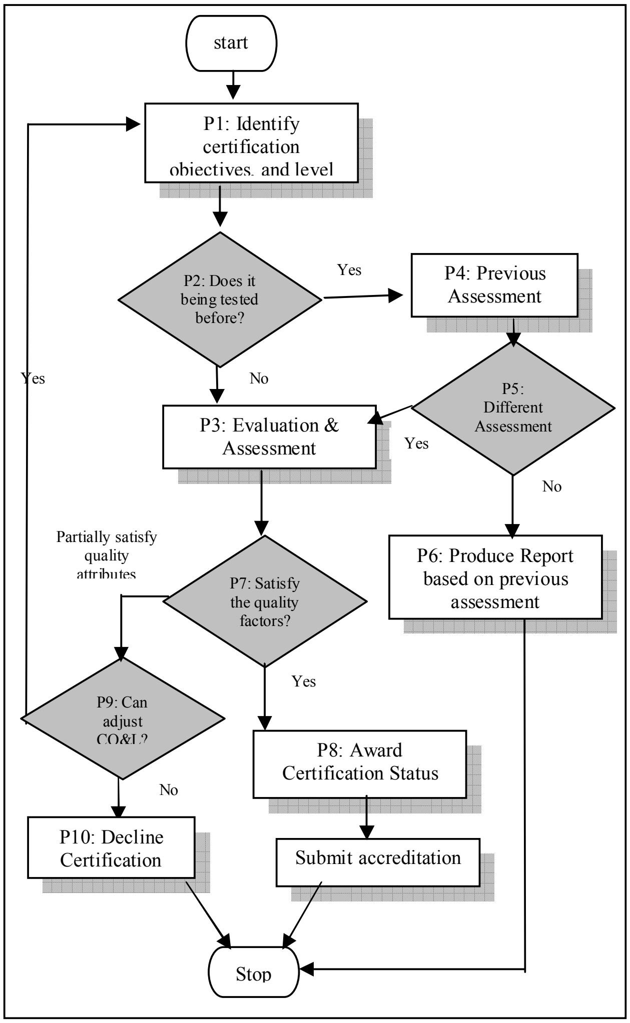

Figure 2. A process decision of Software Certification

In general, the applications of the case studies have demonstrated the feasibility and practicality of the model. On completion of the assessment and certification exercise, the owner of the product was requested to evaluate the model by filling the evaluation and feedback form. The evaluation is to verify the integrity of the model. The owner of the product who is the representative of the organization gives feedback of the assessment and certification results. This feedback and evaluation form is meant to ensure that the owner of the product accepts the results and to verify that the results reflect the actual quality status of the software product. If the owner of the software product does not agree with the results, they may give comments and suggestions in the form. In these case studies, all cases agree with the certification and assessment results and therefore they verify the integrity and validity of the model.

IV. CONCLUSION

The software certification model proposed in this paper has been conceptualized and implementation in three real case studies. The model has been refined and improved by incorporating the contributing factors associated with assessment technique and quality attributes and metrics. The measurements of quality attributes formulated during the design phase and subsequently applied during assessment phase support the model. The application on case studies has illustrated the practicality and feasibility of the proposed process steps and its conceptual model. The evaluation of the model by the software product's owner verified the integrity of the model. It is worth pointing out that the model is definitive, as the results from the application and evaluation process have strongly verified the model.

REFERENCES

[1] J. Voas, "Limited Software Warranties". *Engineering of Computer Based Systems, (ECBS2000) Proceeding, 2000,* pp. 56-61.

[2] J. Voas, "An Approach to certifying Off-the-Shelf Software Components". http://citeseer.ist.psu.edu/voas98approach.html, 1998, (retrieved 30 October 2006).

[3] S. Lauesen, & H. Younessi, "Is Software Quality Visible in the Code?", *IEEE Software,* pp. 69-73, July/August 1998.

[4] G.R. Dromey, "Software Quality - Prevention versus Cure?", *Software Quality Journal* **11:**197-210, 2003.

[5] M.F. Bertoa, J.M. Troya, and A. Vallecillo, "A Survey on the Quality Information Provided by Software Component Vendors", in Proceedings of 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003),2003, pp. 25-30.

[6] J. Voas, "Certification: Reducing the hidden cost of poor quality", *IEEE Software*, pp.22-25, July/August 1999.

[7] J.H. Yahaya, A. Deraman, A.R. Hamdan and A.S. Yahaya, "Software Product Certification Process: The Empirical Study and Conceptual Framework", in *Proceedings of International Conference on Robotics, Vision, Information and Signal Processing, Penang, Malaysia*, pp. 809-813, 2005.

[8] R. Pressman, *Software Engineering A Practitioner's Approach.* McGraw-Hill International Edition, 2001.

[9] Dekkers, C.A. & McQuaid, P.A. The dangers of using software metrics to (Mis)Manage., IT Pro, pp.24-30, March/April 2002.

[10] J.H.Yahaya, A. Deraman & A.R. Hamdan, "Software Quality and Certification: Perception and Practices in Malaysia", *Journal of ICT*, Vol 5, 2006.

[11] Eagles, "Evaluation of Natural Language Processing Systems", Final Report (Sept 1995). http://www.issco.unige.ch/ewg95/ (retrieved 18 May 2007).

[12] M. Ortega, M. Perez, & T. Rojas, "Construction of a systemic Quality Model for Evaluating a software product". *Software Quality Journal* **11:** 219-242, 2003.

[13] J.H. Yahaya, A. Deraman and A.R. Hamdan, "Decision Framework for Software Product Certification Process", in *Proceedings of The Second Malaysian Software Engineering Conference (MySEC'06)* – Kuala Lumpur, Dec 11-12, 2006.

[14] J. H. Yahaya, A. Deraman and A. R. Hamdan. Software product certification model: Classification of quality attributes. *The First Regional Conference of Computational Science and Technology (RCCST 07), Kota Kinabalu,* pp. 436-440, 2007.